AD-A115 667 STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB F/G 12/1
ON THE CONVERGENCE OF A SEQUENTIAL QUADRATIC PROGRAMMING METHOD--ETC(U)
JAN 82 K SCHITTKOWSKI N00014-75-C-0267
UNCLASSIFIED SOL-TR-82-4 ARO-18424.5-MA NL

1 OF 1

AD A

END

DATE

FILMED

7-82

DTIC

ARO 18424.5-MA

Systems
Optimization
Laboratory

AD A115667

DTIC FILE COPY

DTIC
ELECTE
JUN 21 1982
S D
H

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Department of Operations Research
Stanford University
Stanford, CA 94305

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

On The Convergence Of A Sequential Quadratic Programming Method With An Augmented Lagrangian Line Search Functions

by

12

Klaus Schittkowski

TECHNICAL REPORT SOL 82-4

January 1982

Research and reproduction of this report were partially supported by the Department of Energy Contract AM03-76SF00326, PA# DE-AT03-76ER72018; Office of Naval Research Contract N00014-75-C-0267; National Science Foundation Grants MCS-8119774, MCS-7926009 and ECS-8012974; and Army Research Office Contract DAA29-79-C-0110.

ON THE CONVERGENCE OF A SEQUENTIAL QUADRATIC PROGRAMMING METHOD WITH AN AUGMENTED LAGRANGIAN LINE SEARCH FUNCTION

by

Klaus Schittkowski †

Systems Optimization Laboratory
Department of Operations Research
Stanford University
Stanford, CA 94305

January 1982

ABSTRACT

Sequential quadratic programming methods as developed by Wilson, Han, and Powell have gained considerable attention in the last few years mainly because of their outstanding numerical performance. Although the theoretical convergence aspects of this method and its various modifications have been investigated in the literature, there still remain some open questions which will be treated in this paper. The convergence theory to be presented, takes into account the additional variable introduced in the quadratic programming subproblem to avoid inconsistency, the one-dimensional minimization procedure, and, in particular, an "active set" strategy to avoid the recalculation of unnecessary gradients. This paper also contains a detailed mathematical description of a nonlinear programming algorithm which has been implemented by the author. The usage of the code and detailed numerical test results are presented in [15].

This research was supported by the Deutsche Forschungsgemeinschaft while the author was visiting the Systems Optimization Laboratory.

† Institut für Angewandte Mathematik und Statistik, Universität Würzburg, Germany, F.R.

1. Introduction

Consider the general nonlinear optimisation problem

$$\begin{aligned} &\text{minimise} && f(x) \\ &x \in \Re^n: && g_j(x) = 0, \quad j = 1, \ldots, m_e, \\ & && g_j(x) \geq 0, \quad j = m_e + 1, \ldots, m, \end{aligned} \tag{1}$$

with continuously differentiable functions f and g_j, $j = 1, \ldots, m$. One of the most effective tools available today for solving (1), is the sequential quadratic programming algorithm as developed by Wilson [16], Han [6], and Powell [10]. In this method, a line search is performed along a search direction obtained by solving a quadratic programming subproblem. The algorithmic (see Han [5], Powell [11]) and numerical (see [12]) behaviour of the method have been examined and various modifications have been proposed to overcome certain difficulties. For example, the line search procedure may impede superlinear convergence (see Maratos [9]), and the algorithm may cycle (see Chamberlain [2]). One possible remedy (see [13]) is to replace the non-differentiable L_1-line search function used by Han and Powell by a differentiable augmented Lagrange function. However, the convergence analysis of the original method and of the above modification is based on some assumptions which are often not satisfied in practice and there are a few additional numerical drawbacks:

1. All convergence proofs known so far to the author, assume that every quadratic subproblem is feasible. However, this assumption is not always satisfied and Powell [10] proposed the introduction of an additional variable in the subproblem to guarantee consistency. It will be shown that the resulting algorithm will converge if the corresponding penalty parameter is sufficiently large. A lower bound for the choice of this penalty parameter is given.
2. The convergence proof of Han [6] is based on an Armijo-type line search procedure. However, this could lead to an inefficient algorithm and Powell [10] proposed a combination of the Armijo-type line search with a quadratic approximation. This modification leads to a slight alteration of the existing convergence proof.
3. A numerical drawback of the method of Wilson, Han, and Powell is the unnecessary calculation of the gradients of constraints which are inactive at the optimal solution. The experimental tests of [14,15] show that an "active set" strategy can lead to a considerable saving of gradient calculations. It remains to be seen whether it is possible to prove the convergence of the resulting algorithm.
4. The augmented Lagrange function defined in [13] for the line search calculation uses one monotone increasing penalty parameter for all constraints. To improve the robustness of the algorithm, the penalty parameters are now chosen individually for each constraint, their calculation is simplified, and they are allowed to decrease at the beginning of the algorithm.
5. Any sequential quadratic programming algorithm will have difficulties in finding a suitable descent direction for the line search function, if the quadratic subproblem does not satisfy a constraint qualification. A remedy will be proposed in this paper based on an augmented Lagrangian type search direction.

Point 3 mentioned above, is of special importance. One of the basic open questions in nonlinear programming is whether an active set strategy leading to equality constrained subproblems, will be superior to a sequential quadratic programming algorithm with inequality constraints, or vice versa. It is likely that only a combination of both approaches will lead to an efficient, robust, and generally applicable algorithm, and one could consider the proposed "active set" modification as a first approach in finding a suitable compromise.

In Section 2 of the paper, the augmented Lagrangian line search function and the quadratic subproblem are defined. The algorithm is outlined in Section 3 together with some implementation remarks. Section 4 contains the global convergence analysis and further remarks are given in Section 5.

2. Basic concepts

An important tool in nonlinear programming is the Lagrange function

$$L(x,u) = f(x) - \sum_{j=1}^{m} u_j g_j(x) \tag{2}$$

with $x \in \Re^n, u = (u_1, \ldots, u_m)^T \in \Re^m$, which is involved in the well-known necessary optimality conditions, i.e. the Kuhn-Tucker conditions for problem (1)

$$\begin{array}{ll} \text{a)} & \nabla_x L(x,u) = 0, \\ \text{b)} & g_j(x) = 0, \quad j = 1,\ldots,m_e, \\ \text{c)} & g_j(x) \geq 0, \quad j = m_e+1,\ldots,m, \\ \text{d)} & u_j \geq 0, \quad j = m_e+1,\ldots,m, \\ \text{e)} & g_j(x)u_j = 0, \quad j = m_e+1,\ldots,m. \end{array} \tag{3}$$

Here, ∇_x denotes differentiation with respect to the x-variables. A sequential quadratic programming algorithm proceeds from a quadratic approximation of the Lagrange function (2) and a linearisation of the constraints. If x_k denotes the k-th estimate for the optimal solution and B_k a symmetric matrix that approximates the Hessian of the Lagrange function, the resulting quadratic programming subproblem can be written in the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d \\ d \in \Re^n: & \nabla g_j(x_k)^T d + g_j(x_k) = 0, \quad j = 1,\ldots,m_e, \\ & \nabla g_j(x_k)^T d + g_j(x_k) \geq 0, \quad j = m_e+1,\ldots,m. \end{array} \tag{4}$$

The next iterate is given by

$$x_{k+1} = x_k + \alpha_k d_k,$$

where d_k denotes the solution of (4) and α_k a steplength parameter which will be discussed later. A numerical drawback of using (4) is that all gradients of the constraints must be evaluated in each iteration step, even if x_k is close to the solution and we can suppose therefore that the calculation of inactive nonlinear constraints is unnecessary. This statement is at least true if we expect that nonlinear constraints inactive at the optimal solution, correspond to linearised constraints inactive at a solution of (4). To avoid this situation and to improve the efficiency of the algorithm, an alternative subproblem may be defined as follows:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d \\ d \in \Re^n: & \nabla g_j(x_k)^T d + g_j(x_k) \left\{ {= \atop \geq} \right\} 0, \quad j \in J_k^*, \\ & \nabla g_j(x_{k(j)})^T d + g_j(x_k) \geq 0, \quad j \in K_k^*. \end{array} \tag{5}$$

J_k^* and K_k^* are two disjoint index sets with $J_k^* \cup K_k^* = \{1,\ldots,m\}$. J_k^* is called the set of the active constraints including the equality constraints, and K_k^* is the set of the inactive constraints. The indices $k(j) \leq k$ correspond to previous iterates and their definition will be clear when investigating the algorithm.

To motivate the choice of the active set J_k^*, we observe that the algorithm approximates not only the optimal solution by x_k, but also the optimal Lagrange multipliers. The variables corresponding to the Lagrange multipliers, are denoted by $v_k = (v_1^{(k)}, \ldots, v_m^{(k)})^T$. A constraint is called active, i.e. its index is in J_k^*, if its function value is not positive or if the corresponding multiplier is greater than zero. Given a constant $\epsilon > 0$ and any iterates x_k and v_k, we set

$$\begin{aligned} J_k^* &\doteq \{1,\ldots,m_e\} \cup \{j : m_e < j \le m, \quad g_j(x_k) \le \epsilon \quad \text{or} \quad v_j^{(k)} > 0\}, \\ K_k^* &\doteq \{1,\ldots,m\} \setminus J_k^*. \end{aligned} \tag{6}$$

If x_k is feasible, v_k is replaced by the optimal Lagrange multiplier of (1), and ϵ sufficiently small, then J_k^* defines the constraints which are active at the optimal solution of (1). By using the condition $g_j(x_k) \le \epsilon$ instead of $g_j(x_k) \le 0$, we attempt to avoid the situation in which $g_j(x_k)$ tends to zero for $j \in K_k^*$.

However, the linear constraints in (4) or (5) can become inconsistent even if we assume that the original problem (1) is solvable. As in Powell [10], an additional variable δ is introduced in (5), leading to an $(n+1)$-dimensional subproblem with consistent constraints:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d + \frac{1}{2}\rho_k \delta^2 \\ d \in \Re^n, \delta \in \Re : & \nabla g_j(x_k)^T d + (1-\delta) g_j(x_k) \left\{ \begin{smallmatrix} = \\ \geq \end{smallmatrix} \right\} 0, \quad j \in J_k^*, \\ & \nabla g_j(x_{k(j)})^T d + g_j(x_k) \ge 0, \quad j \in K_k^*, \\ & 0 \le \delta \le 1. \end{array} \tag{7}$$

Obviously, the point $d_0 = 0$, $\delta_0 = 1$ satisfies the constraints of (7), since $g_j(x_k) > 0$ for all $j \in K_k^*$. We conclude that (7) has a finite unique solution provided that the matrix B_k is positive-definite. The additional penalty parameter ρ_k can be chosen by

$$\rho_k \doteq \max\left(\rho_0 \, , \, \frac{\rho^* (d_{k-1}^T A_{k-1} u_{k-1})^2}{(1-\delta_{k-1})^2 d_{k-1}^T B_{k-1} d_{k-1}} \right) \tag{8}$$

for $k > 0$ and a constant $\rho^* \ge 1$. A motivation for this rule is given by the convergence analysis of Section 4. Here, A_{k-1} denotes the matrix

$$A_{k-1} \doteq (\nabla g_1(x_{k-1}), \ldots, \nabla g_m(x_{k-1})).$$

Now assume that we have succeeded in solving (7), giving us a search direction d_k and a multiplier $u_k = (u_1^{(k)}, \ldots, u_m^{(k)})^T$. As mentioned above, the variables and the multipliers are updated simultaneously by

$$x_{k+1} \doteq x_k + \alpha_k d_k, \quad v_{k+1} \doteq v_k + \alpha_k (u_k - v_k).$$

The steplength parameter α_k is obtained by minimising a line search function or merit function. Han [6] and Powell [10] used the non-differentiable L_1- penalty function

$$P(x,r) \doteq f(x) + \sum_{j=1}^{m_e} r_j \mid g_j(x) \mid + \sum_{j=m_e+1}^{m} r_j \mid \min(0, g_j(x)) \mid \tag{9}$$

with $r = (r_1, \ldots, r_m)^T$. The use of (9) alone can lead to two difficulties. First, the superlinear convergence may be impeded even in an arbitrarily small neighbourhood of the solution (see Maratos [9]). Second, the algorithm may cycle if the penalty parameters are chosen improperly (see Chamberlain [2]). To overcome these situations, (9) has been replaced by the differentiable augmented Lagrange function in [13], i.e. by

$$\phi_r(x,v) \doteq f(x) - \sum_{j\in J}(v_j g_j(x) - \tfrac{1}{2}r_j g_j(x)^2) - \tfrac{1}{2}\sum_{j\in K} v_j^2/r_j. \tag{10}$$

Here, the index sets J and K are defined by

$$\begin{aligned} J &\doteq \{1,\ldots,m_e\} \cup \{j : m_e < j \le m, \quad g_j(x) \le v_j/r_j\}, \\ K &\doteq \{1,\ldots,m\} \setminus J. \end{aligned} \tag{11}$$

However, we must be very careful when replacing (9) by (10) in an optimisation algorithm. The difficulty arises that a solution of (1) is only a saddle point of the function ϕ_r with respect to the variables (x,v). In other words, a formulation of an optimisation algorithm as a descent method for ϕ_r with a constant penalty parameter r can lead to a sequence (x_k, v_k) that tends to infinity, even if the feasible region of (1) is compact. To avoid this undesirable behavior of the algorithm, the penalty parameter in (10) must be adapted in an appropriate way and is defined by

$$\begin{aligned} r_j^{(k+1)} &\doteq \max\left(\sigma_j^{(k)} r_j^{(k)}, \frac{2m(u_j^{(k)} - v_j^{(k)})^2}{(1-\delta_k)d_k^T B_k d_k}\right), \quad j = 1,\ldots,m, \\ r_{k+1} &\doteq (r_1^{(k+1)},\ldots,r_m^{(k+1)})^T, \end{aligned} \tag{12}$$

where (d_k, u_k) is a Kuhn-Tucker point of the quadratic subproblem, δ_k the additional variable from (7) ($\delta_k \ne 1$), r_k the previous penalty parameter, v_k the current approximation of the multiplier vector, and B_k a positive-definite approximation of the Hessian of the Lagrange function. The function

$$\varphi_k(\alpha) \doteq \phi_{r_{k+1}}\left(\begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}\right), \tag{13}$$

with

$$\begin{aligned} J_k &\doteq \{1,\ldots,m_e\} \cup \{j : m_e < j \le m, \quad g_j(x_k) \le v_j^{(k)}/r_j^{(k+1)}\}, \\ K_k &\doteq \{1,\ldots,m\} \setminus J_k, \end{aligned} \tag{14}$$

can be minimized with respect to α, leading to a steplength α_k. We must distinguish between the index sets J_k^*, cf. (7), and J_k, cf. (14), which are both approximations of the optimal active set of (1). It is easy to see that

$$J_k^* \supseteq J_k, \quad K_k^* \subseteq K_k. \tag{15}$$

The sequence $\{\sigma_j^{(k)}\}$ is included in (12) to allow for decreasing penalty parameters at least in the beginning of the algorithm, if we require that $\sigma_j^{(k)} \le 1$. On the other hand, it should guarantee the convergence of $\{r_j^{(k)}\}$ whenever this sequence is bounded. A sufficient condition is given by the following lemma.

(2.1)Lemma: *Assume that $\{r_j^{(k)}\}_{k\in\mathbb{N}}$ is bounded, $\sigma_j^{(k)} \le 1$ for all k, and that*

$$\sum_{k=0}^{\infty}(1-\sigma_j^{(k)}) < \infty, \quad 1 \le j \le m.$$

Then there is a $r_j^* \ge 0$ with

$$\lim_{k\to\infty} r_j^{(k)} = r_j^*.$$

Proof: To simplify the notation, we omit the index j and define R as the upper bound of the penalty parameters. Assume that there are two different accumulation points r^* and r^{**} of $\{r^{(k)}\}$ with $r^* < r^{**}$. Then we obtain for each $\epsilon > 0$ infinitely many k and l_k with

$$| r^{(k+l_k)} - r^* | \leq \epsilon, \quad | r^{(k)} - r^{**} | \leq \epsilon.$$

Setting $l \doteq l_k$ and choosing a sufficiently small $\epsilon > 0$, we obtain

$$\begin{aligned} 0 &< r^{**} - r^* - 2\epsilon \\ &\leq -(r^{(k+l)} - r^{(k)}) \\ &\leq R \sum_{i=0}^{l-1} (1 - \sigma^{(k+i)}). \end{aligned}$$

Since this inequality is valid for infinitely many k and the right-hand side tends to zero, we get a contradiction. ∎

A possible choice of $\sigma_j^{(k)}$ could be

$$\sigma_j^{(k)} \doteq 1 - \left(1 - \frac{1}{\sqrt{r_j^{(k)}}}\right)^k.$$

Since we expect that only large penalty parameters could affect the performance of the algorithm, we can replace the above formula by a simpler approximation

$$\sigma_j^{(k)} \doteq \min\left(1\, , \, \frac{k}{\sqrt{r_j^{(k)}}}\right). \tag{16}$$

3. The algorithm

Now we are able to formulate the algorithm. First, some constants ϵ, β, μ, $\bar{\delta}$, $\bar{\rho}$ have to be chosen that are not changed within the algorithm and that satisfy

$$\epsilon > 0, \quad 0 < \beta < 1, \quad 0 < \mu < \tfrac{1}{2}, \quad 0 < \bar{\delta} < 1, \quad \bar{\rho} > 1. \tag{17}$$

The main steps consist of the following instructions:

(3.1) Algorithm :

0) Start: Choose some starting values $x_0 \in \Re^n$, $v_0 \in \Re^m$, $B_0 \in \Re^n \times \Re^n$ positive-definite, $\rho_0 \in \Re$, $r_0 \in \Re^m$, and evaluate $f(x_0)$, $g_j(x_0)$, $j = 1,\dots,m$, $\nabla f(x_0)$, $\nabla g_j(x_0)$, $j = 1,\dots,m$. Determine J_0^* and let $k(j) = 0$ for all $j \in K_0^*$.

For $k = 0,1,2,\dots$ compute x_{k+1}, v_{k+1}, B_{k+1}, r_{k+1}, ρ_{k+1}, and J_{k+1}^* as follows:

1) Solve the quadratic subproblem (7) and denote by d_k, δ_k the optimal solution and by u_k the optimal multiplier. If $\delta_k > \bar{\delta}$, let $\rho_k \doteq \bar{\rho}\rho_k$ and solve (7) again. If this loop fails within a given upper bound for δ_k, define

$$\begin{aligned} d_k &\doteq -B_k^{-1}\nabla_x \phi_{r_k}(x_k, v_k), \\ u_k &\doteq v_k - \nabla_v \phi_{r_k}(x_k, v_k). \end{aligned} \tag{18}$$

2) Determine the new penalty parameter r_{k+1} by (12) and (16). If d_k , u_k have been obtained by (18), let $r_{k+1} = r_k$.

3) If $\varphi_k'(0) \geq 0$, let $\rho_k \doteq \bar{\rho}\rho_k$ and go to 1).

4) Define the new penalty parameter ρ_{k+1} by (8).

5) Perform a line search with respect to the function $\varphi_k(\alpha)$ defined by (13). Let $\alpha_{k,0} \doteq 1$ and, for $i = 1,2,\dots$, let i_k be the first index for which

$$\varphi_k(\alpha_{k,i}) \leq \varphi_k(0) + \mu\alpha_{k,i}\varphi_k'(0), \tag{19}$$

where $\alpha_{k,i} = \max(\beta\alpha_{k,i-1}, \bar{\alpha}_{k,i-1})$. Here, $\bar{\alpha}_{k,i-1}$ is the minimiser of a quadratic approximation of $\varphi_k(\alpha)$ using $\varphi_k(0)$, $\varphi_k'(0)$, and $\varphi_k(\alpha_{k,i-1})$. Define

$$\alpha_k \doteq \alpha_{k,i_k}.$$

6) Let

$$\begin{aligned} x_{k+1} &\doteq x_k + \alpha_k d_k, \\ v_{k+1} &\doteq v_k + \alpha_k(u_k - v_k), \end{aligned}$$

and evaluate $f(x_{k+1})$, $g_j(x_{k+1})$, $j = 1,\dots,m$, $\nabla f(x_{k+1})$, J_{k+1}^* by (6), and $\nabla g_j(x_{k+1})$, $j \in J_{k+1}^*$.

7) Compute a suitable new positive-definite approximation of the Hessian of the Lagrange function, i.e. B_{k+1}, set $k \doteq k+1$, and repeat the iteration.

The following remarks will illustrate further details of the algorithm. Numerical experience shows that the definition of the parameters satisfying (17) is not a crucial part for the performance. Suitable values are

$$\epsilon = 10^{-7}, \quad \beta = .1, \quad \mu = .1, \quad \bar{\delta} = .9, \quad \bar{\rho} = 100.$$

A stopping criterion has been omitted to facilitate the description of the algorithm. For a suitable condition, one could use Powell's [10] proposal or any other rules, for example

$$d_k^T B_k d_k \leq \epsilon^2,$$

$$\sum_{j=1}^{m} | u_j^{(k)} g_j(x_k) | \leq \epsilon,$$

$$\| \nabla_x L(x_k, u_k) \|^2 \leq \epsilon,$$

$$\sum_{j=1}^{m_e} | g_j(x_k) | + \sum_{j=m_e+1}^{m} | \min(0, g_j(x_k)) | \leq \sqrt{\epsilon}.$$

The corresponding tolerance ϵ, which must be provided by the user, could also be applied to define the active set J_k^*, cf. (6). Here, ϵ should be sufficiently small so that

$$0 < \epsilon \leq g_j(x^*)$$

for all $j > m_e$ with $g_j(x^*) > 0$ and for the optimal solution x^*. If ϵ is chosen too large, the only disadvantage is that some additional gradient evaluations are required.

A user often has a suitable guess for the starting point x_0. If nothing is known about the multiplier and the Hessian of the Lagrangian, one could define $v_0 \doteq 0$, $B_0 \doteq I$, and one could set $\rho_0 \doteq 1$, $r_j^{(0)} \doteq 1$, $j = 1, \ldots, m$, or even larger, if a numerically stable algorithm for solving the quadratic subproblem is available.

Numerical tests indicate that the penalty parameter ρ_k in (7) could influence the performance of Algorithm (3.1). For this reason, the numerical implementation [15] contains an additional option to solve the quadratic subproblem (5) first, and to proceed to (7) only if (5) turns out to be infeasible. Note that the convergence results of Section 4 remain valid if this option is preferred. Furthermore, the choice of ρ_k by (8) is adapted to the current state of the algorithm to avoid unnecessarily ill-conditioned matrices of the form

$$\begin{pmatrix} B_k & 0 \\ 0 & \rho_k \end{pmatrix}$$

in the quadratic programming routine.

The loop in Step 1) of Algorithm (3.1) could fail only if the subproblem does not satisfy a constraint qualification. In this case, the modified search direction (18) is used with the intention of minimizing the augmented Lagrange function ϕ_{r_k}. The loop between Step 3) and Step 1) is finite, since a lower bound for the choice of ρ_k can be given, cf. Section 4.

When solving the subproblem (7) by any "black box" quadratic programming subroutine, one overlooks the fact that in a quasi-Newton implementation, the matrix B_k is updated by only a rank-two correction. To improve the numerical efficiency of the algorithm, in particular, if only a few constraints are active, one could use a Cholesky factorisation of B_k. For a description of the corresponding LDL-factors see [14]. Then the quadratic subproblem is identical with a least-squares problem which could, for example, be solved by the programs published in Lawson and Hanson [8].

The definition of the penalty parameter r_{k+1} is closely related to the algorithm presented in [13]. However, there are two differences. First, the penalty parameters are chosen individually for each constraint, and second, bounded parameters are not expected to be constant as in [13], so that the resulting algorithm should be more efficient and robust. Nevertheless it is possible that r_{k+1} tends to infinity. The convergence analysis of Section 4 will show that in this case, the convergence of Algorithm (3.1) can be proved without using a line search, which indicates that this situation should occur rarely in practical situations. The specific choice of r_{k+1} is motivated by the convergence requirement to generate descent directions for the augmented Lagrange function $\phi_{r_{k+1}}$. The parameter r_{k+1} will be large when the improvement in the approximation of the variables, i.e. d_k, will be smaller than the improvement in the approximation of the multipliers, i.e. $u_k - v_k$.

The line search procedure of Step 5) in Algorithm (3.1) is a very simple method and is justified by the excellent numerical results obtained with the original implementation of Powell, cf. [12], and in further tests, cf. [14,15]. It is expected that only for badly scaled problems, this procedure should be replaced by a more sophisticated algorithm, cf. for example Gill, Murray, and Wright [3]. A straightforward classroom calculation shows that the quadratic approximation of $\varphi_k(\alpha)$ is minimized by the expression

$$\bar{\alpha}_{k,i} = \frac{1}{2}\frac{\alpha_{k,i}^2 \varphi_k'(0)}{\alpha_{k,i}\varphi_k'(0) - (\varphi_k(\alpha_{k,i}) - \varphi_k(0))} \tag{20}$$

with

$$\varphi_k'(0) = \nabla \phi_{r_{k+1}}(x_k, v_k)^T \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}.$$

It will be shown in Section 4 that $\varphi_k'(0) < 0$, and that the line search algorithm is finite.

When investigating Step 6) of Algorithm (3.1), the choice of the variables $x_{k(j)}$ in (6) can be explained. In the matrix defining the linear constraints of the subproblem, only those rows are replaced in the k-th iteration step, for which $j \in J_{k+1}^*$. The others remain as the previously computed gradients.

Finally, a suitable approximation of the Hessian of the Lagrangian must be found. The extensive numerical experience gathered in recent years shows that this Hessian can be approximated by a variable metric formula with positive-definite matrices B_k, even if the true Hessian of the Lagrange function is indefinite. Since excellent numerical results are obtained with Powell's modification of the BFGS-formula, cf. [12], the usage of this formula or its equivalent inverse formulation, if one wants to avoid the inversion of triangular factors, is recommended. For more information about this variable metric formula, see Powell [10] or [14] for the definition of the corresponding LDL-factors.

4. Global convergence analysis

The convergence analysis of Algorithm (3.1) depends mainly on the Kuhn-Tucker conditions for the quadratic programming subproblem (7) which can be written in the following form:

$$
\begin{aligned}
&\text{a)} \quad B_k d_k + \nabla f(x_k) - \sum_{j \in J_k^*} u_j^{(k)} \nabla g_j(x_k) - \sum_{j \in K_k^*} u_j^{(k)} \nabla g_j(x_{k(j)}) = 0, \\
&\text{b)} \quad \rho_k \delta_k + \sum_{j \in J_k^*} u_j^{(k)} g_j(x_k) - \nu_1^{(k)} + \nu_2^{(k)} = 0, \\
&\text{c)} \quad w_j^{(k)} = 0, \quad j = 1, \ldots, m_e, \\
&\text{d)} \quad w_j^{(k)} \geq 0, \quad j = m_e + 1, \ldots, m, \\
&\text{e)} \quad 0 \leq \delta_k \leq 1, \\
&\text{f)} \quad u_j^{(k)} \geq 0, \quad j = m_e + 1, \ldots, m, \\
&\text{g)} \quad \nu_1^{(k)} \geq 0, \\
&\text{h)} \quad \nu_2^{(k)} \geq 0, \\
&\text{i)} \quad w_j^{(k)} u_j^{(k)} = 0, \quad j = 1, \ldots, m, \\
&\text{j)} \quad \nu_1^{(k)} \delta_k = 0, \\
&\text{k)} \quad \nu_2^{(k)} (1 - \delta_k) = 0,
\end{aligned}
\tag{21}
$$

where

$$
\begin{aligned}
w_j^{(k)} &\doteq \nabla g_j(x_k)^T d_k + (1 - \delta_k) g_j(x_k), \quad j \in J_k^*, \\
w_j^{(k)} &\doteq \nabla g_j(x_{k(j)})^T d_k + g_j(x_k), \quad j \in K_k^*.
\end{aligned}
\tag{22}
$$

$\nu_1^{(k)}$ and $\nu_2^{(k)}$ are the multipliers with respect to the lower and upper bounds for the additional variable δ.

First we have to investigate whether Algorithm (3.1) is well defined and start with considering the internal loop of Step 1).

(4.1) Lemma: *Assume that (7) satisfies the constraint qualification, i.e. that the gradients of the constraints active at the optimal solution are independent, and that the feasible region of (7) is bounded for each k. Then the loop in Step 1) of Algorithm (3.1) is finite.*

Proof: To simplify the proof, we omit the iteration index k and assume that there are infinitely many ρ_i with $\lim_{i\to\infty} \rho_i = \infty$, each giving a solution d_i, δ_i of (7) and a multiplier u_i. Since $\delta_i \bar{\delta} > 0$, we obtain from (21b)

$$
0 \geq \rho_i \delta_i + \sum_{j \in J^*} u_j^{(i)} g_j(x) \geq \rho_i \bar{\delta} + \sum_{j \in J^*} u_j^{(i)} g_j(x)
$$

indicating that $\lim_{i\to\infty} \| u_i \| = \infty$. If $\bar{u}_i$ denotes the non-zero part of u_i and $\bar{A}_i$ the matrix consisting of the corresponding gradients ∇g_j, we write (21a) in the form

$$
B d_i + \nabla f(x) - \bar{A}_i \bar{u}_i = 0
$$

or

$$
\bar{u}_i = (\bar{A}_i^T \bar{A}_i)^{-1} \bar{A}_i^T (B d_i + \nabla f(x)).
$$

Since, however, $\{d_i\}$ is bounded, we obtain a contradiction. ∎

The boundedness of the feasible region in (7) will henceforth be assumed now in the further global convergence analysis. The iterates x_k can be forced to remain bounded, if additional lower and upper bounds are given in (1), i.e. if there are $x_l, x_u \in \Re^n$ with

$$x_l \leq x \leq x_u, \tag{23}$$

also implying the boundedness of $\{d_k\}$ provided that $\{\alpha_k\}$ does not approximate zero.

The subsequent theorem will be fundamental for the convergence analysis. It shows that the computed search direction is a descent direction for $\phi_{r_{k+1}}$, i.e. that the line search is well defined, and that it leads to a sufficiently large decrease of $\phi_{r_{k+1}}$. First, some notation will be introduced to facilitate the proof. If x_k, v_k, J_k are some iterates of Algorithm (3.1) and r_{k+1} the corresponding penalty parameter, we set

$$\begin{aligned}
\bar{v}_k &\doteq (\bar{v}_1^{(k)}, \ldots, \bar{v}_m^{(k)})^T, & \bar{v}_j^{(k)} &= \begin{cases} v_j^{(k)}, & \text{if } j \in J_k, \\ 0, & \text{otherwise} \end{cases} \\
\bar{w}_k &\doteq (\bar{w}_1^{(k)}, \ldots, \bar{w}_m^{(k)})^T, & \bar{w}_j^{(k)} &= \begin{cases} w_j^{(k)}, & \text{if } j \in J_k, \\ 0, & \text{otherwise ,} \end{cases} \\
g_k &\doteq (g_1(x_k), \ldots, g_m(x_k))^T, & & \\
\bar{g}_k &\doteq (\bar{g}_1(x_k), \ldots, \bar{g}_m(x_k))^T, & \bar{g}_j(x_k) &= \begin{cases} g_j(x_k), & \text{if } j \in J_k, \\ 0, & \text{otherwise ,} \end{cases} \\
\bar{g}'_k &\doteq (\bar{g}'_1(x_k), \ldots, \bar{g}'_m(x_k))^T, & \bar{g}'_j(x_k) &= \begin{cases} g_j(x_k), & \text{if } j \in J_k, \\ v_j^{(k)}/r_j^{(k)}, & \text{otherwise ,} \end{cases} \\
A_k &\doteq (\nabla g_1(x_k), \ldots, \nabla g_m(x_k)), & & \\
R_{k+1} &\doteq \operatorname{diag}(r_1^{(k+1)}, \ldots, r_m^{(k+1)}) & &
\end{aligned} \tag{24}$$

Then we can express the gradient of $\phi_{r_{k+1}}(x_k, v_k)$ in the following form:

$$\nabla \phi_{r_{k+1}}(x_k, v_k) = \begin{pmatrix} \nabla f(x_k) - A_k(\bar{v}_k - R_{k+1}\bar{g}_k) \\ -\bar{g}'_k \end{pmatrix}. \tag{25}$$

(4.2) Theorem: *Let x_k, v_k, d_k, δ_k, u_k, B_k, r_k, ρ_k, and J_k^* be given iterates of Algorithm (3.1), $k \geq 0$, and assume that*

(i) $d_k^T B_k d_k \geq \gamma \parallel d_k \parallel^2$ *for a $\gamma \in \Re$ with $0 < \gamma \leq 1$,*

(ii) $\delta_k \leq \bar{\delta}$,

(iii) $\rho_k \geq \dfrac{\parallel A_k \bar{v}_k \parallel^2}{\gamma(1-\bar{\delta})^2}$.

Then

$$\nabla \phi_{r_{k+1}}(x_k, v_k)^T \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix} \leq -\tfrac{1}{4}\gamma \parallel d_k \parallel^2 . \tag{26}$$

Proof: We use the Kuhn-Tucker conditions (21) for the quadratic subproblem and get (26) by the following estimates, where we omit the iteration index k:

$$
\begin{aligned}
&-\nabla\phi_r(x,v)^T\begin{pmatrix} d \\ u-v \end{pmatrix} \\
&\quad = -\nabla f(x)^T d + d^T A(\bar{v} - R\bar{g}) + \bar{g}^T(u-v) \\
&\quad = d^T Bd - \sum_{j\in J^*} u_j \nabla g_j(x)^T d - \sum_{j\in K^*} u_j \nabla g_j(x_{k(j)})^T d \\
&\qquad + \sum_{j\in J}(v_j - r_j g_j(x))\nabla g_j(x)^T d + \sum_{j\in J} g_j(x)(u_j - v_j) \\
&\qquad + \sum_{j\in K}\frac{1}{r_j} v_j(u_j - v_j) && \text{cf. (21a),(24)} \\
&\quad = d^T Bd - \sum_{j\in J^*}(u_j - v_j)\nabla g_j(x)^T d - \sum_{j\in K^*} u_j g_j(x_{k(j)})^T d \\
&\qquad - \sum_{j\in J^*\setminus J} v_j \nabla g_j(x)^T d - \sum_{j\in J} r_j g_j(x)\nabla g_j(x)^T d \\
&\qquad + \sum_{j\in J}(u_j - v_j) g_j(x) + \sum_{j\in K}\frac{1}{r_j} v_j(u_j - v_j) && \text{cf.(15)} \\
&\quad = d^T Bd - \sum_{j\in J^*}(u_j - v_j) w_j + (1-\delta)\sum_{j\in J^*}(u_j - v_j) g_j(x) \\
&\qquad - \sum_{j\in K^*} u_j w_j + \sum_{j\in K^*} u_j g_j(x) - \sum_{j\in J^*\setminus J} v_j w_j \\
&\qquad + (1-\delta)\sum_{j\in J^*\setminus J} v_j g_j(x) - \sum_{j\in J} r_j g_j(x) w_j + (1-\delta)\sum_{j\in J} r_j g_j(x)^2 \\
&\qquad + \sum_{j\in J}(u_j - v_j) g_j(x) + \sum_{j\in K}\frac{1}{r_j} v_j(u_j - v_j) && \text{cf.(22)} \\
&\quad = d^T Bd + \sum_{j\in J} w_j(v_j - r_j g_j(x)) + \sum_{j\in J^*\setminus J} w_j v_j + 2\sum_{j\in J}(u_j - v_j) g_j(x) \\
&\qquad + \sum_{j\in J^*\setminus J}(u_j - v_j) g_j(x) - \delta\sum_{j\in J^*}(u_j - v_j) g_j(x) + \sum_{j\in K^*} u_j g_j(x) \\
&\qquad - \sum_{j\in J^*\setminus J} v_j w_j + (1-\delta)\sum_{j\in J^*\setminus J} v_j g_j(x) + (1-\delta)\sum_{j\in J} r_j g_j(x)^2 \\
&\qquad + \sum_{j\in K}\frac{1}{r_j} v_j(u_j - v_j) && \text{cf.(21i)} \\
&\quad \geq d^T Bd + 2\sum_{j\in J}(u_j - v_j) g_j(x) + 2\sum_{j\in K}\frac{1}{r_j} v_j(u_j - v_j) \\
&\qquad - \sum_{j\in K}\frac{1}{r_j} v_j(u_j - v_j) + \sum_{j\in J^*\setminus J} u_j g_j(x) - \delta\sum_{j\in J^*}(u_j - v_j) g_j(x) \\
&\qquad + \sum_{j\in K^*} u_j g_j(x) + (1-\delta)\sum_{j\in J} r_j g_j(x)^2 \\
&\qquad - \delta\sum_{j\in J^*\setminus J} v_j g_j(x) && \text{cf.(14),(21d)}
\end{aligned}
$$

$$
\begin{aligned}
&= d^TBd + 2\bar{g}'^T(u-v) - \sum_{j\in K^*}\frac{1}{r_j}v_j(u_j-v_j) - \sum_{j\in K\setminus K^*}\frac{1}{r_j}v_j(u_j-v_j)\\
&\quad + \sum_{j\in K\setminus K^*}u_jg_j(x) - \delta\sum_{j\in J^*\setminus J}u_jg_j(x) - \delta\sum_{j\in J}(u_j-v_j)g_j(x)\\
&\quad + \sum_{j\in K^*}u_jg_j(x) + (1-\delta)\sum_{j\in J}r_jg_j(x)^2 && \text{cf.(15)}\\
&= d^TBd + 2\bar{g}'^T(u-v) + \sum_{j\in K^*}u_j(g_j(x)-\frac{1}{r_j}v_j) + \sum_{j\in K^*}\frac{1}{r_j}v_j^2\\
&\quad + \sum_{j\in K\setminus K^*}u_j(g_j(x)-\frac{1}{r_j}v_j) + \sum_{j\in K\setminus K^*}\frac{1}{r_j}v_j^2 - \delta\sum_{j\in J^*\setminus J}u_jg_j(x)\\
&\quad - \delta\sum_{j\in J}(u_j-v_j)g_j(x) + (1-\delta)\sum_{j\in J}r_jg_j(x)^2\\
&= d^TBd + 2\bar{g}'^T(u-v) + (1-\delta)\bar{g}^TR\bar{g} + \sum_{j\in K}u_j(g_j(x)-\frac{1}{r_j}v_j)\\
&\quad + \sum_{j\in K}\frac{1}{r_j}v_j^2 - \delta\sum_{j\in J^*\setminus J}u_jg_j(x) - \delta\sum_{j\in J}(u_j-v_j)g_j(x)\\
&\geq d^TBd + 2\bar{g}'^T(u-v) + (1-\delta)\bar{g}^TR\bar{g} + (1-\delta)\sum_{j\in K}\frac{1}{r_j}v_j^2\\
&\quad - \delta\sum_{j\in J^*}u_jg_j(x) + \delta\sum_{j\in J}v_jg_j(x) && \text{cf.(21f),(14)}\\
&\geq d^TBd + 2\bar{g}'^T(u-v) + (1-\delta)\bar{g}'^TR\bar{g}' + \rho\delta^2\\
&\quad - \nu_1\delta + \delta\bar{v}^T\bar{g} - \frac{\delta}{1-\delta}\bar{w}^T\bar{v} && \text{cf.(21b,d)}\\
&= d^TBd + \| \sqrt{1-\delta}R^{1/2}\bar{g}' + \frac{1}{\sqrt{1-\delta}}R^{-1/2}(u-v) \|^2\\
&\quad - \frac{1}{1-\delta}(u-v)^TR^{-1}(u-v) + \rho\delta^2\\
&\quad + \delta(\bar{v}^T\bar{g} - \frac{1}{1-\delta}\bar{w}^T\bar{v}) && \text{cf.(21j)}\\
&\geq d^TBd - \frac{1}{1-\delta}(u-v)^TR^{-1}(u-v)\\
&\quad + (\sqrt{\rho}\delta + \frac{1}{2\sqrt{\rho}}(\bar{v}^T\bar{g} - \frac{1}{1-\delta}\bar{w}^T\bar{v}))^2 - \frac{1}{4\rho}(\bar{v}^T\bar{g} - \frac{1}{1-\delta}\bar{w}^T\bar{v})^2\\
&\geq \tfrac{1}{2}d^TBd + \tfrac{1}{2}d^TBd - \frac{1}{1-\delta}(u-v)^TR^{-1}(u-v)\\
&\quad - \frac{1}{4\rho}(\bar{v}^T\bar{g} - \frac{1}{1-\delta}\bar{w}^T\bar{v})^2
\end{aligned}
$$

$$\geq \tfrac{1}{2}\gamma \| d \|^2 + \tfrac{1}{2} d^T B d - \frac{1}{1-\delta} \sum_{j=1}^{m} \frac{1}{r_j}(u_j - v_j)^2$$

$$- \frac{1}{4\rho}(\bar{v}^T(\bar{g} - \frac{1}{1-\delta}\bar{w}))^2 \qquad \text{cf.(i)}$$

$$\geq \tfrac{1}{2}\gamma \| d \|^2 + \tfrac{1}{2} d^T B d - \frac{1}{1-\delta} \sum_{j=1}^{m} \frac{(1-\delta) d^T B d}{2m(u_j - v_j)^2}(u_j - v_j)^2$$

$$- \frac{1}{4\rho(1-\delta)^2}(\bar{v}^T((1-\delta)\bar{g} - \bar{w}))^2 \qquad \text{cf.(12)}$$

$$= \tfrac{1}{2}\gamma \| d \|^2 - \frac{1}{4\rho(1-\delta)^2}(\bar{v}^T A^T d)^2 \qquad \text{cf.(22)}$$

$$\geq \tfrac{1}{2}\gamma \| d \|^2 - \frac{\gamma(1-\delta)^2}{4(1-\delta)^2 \| A\bar{v} \|^2} \| A\bar{v} \|^2 \| d \|^2 \qquad \text{cf.(iii)}$$

$$\geq \tfrac{1}{4}\gamma \| d \|^2 . \qquad \text{cf.(i),(ii)}$$

During the proof we used $v_j^{(k)} \geq 0$ for all $j > m_e$, since $\alpha_{k-1} \leq 1$, and we set

$$R^{1/2} \doteq \mathrm{diag}(\sqrt{r_1}, \ldots, \sqrt{r_m}).$$

∎

Assumption (i) can be considered as a standard assumption henceforth required in the theory of quasi-Newton algorithms. It can be forced by choosing a γ and performing a restart with $B_k = I$ whenever (i) is violated. The validity of assumption (ii) is guaranteed by Step 1) of Algorithm (3.1), since Lemma (4.1) shows that after finitely many sub-iterations, the condition $\delta_k \leq \bar{\delta}$ will be achieved at least under a constraint qualification. Otherwise, d_k and u_k define a descent direction for ϕ_{k+1} in the case when they are replaced by (18). To avoid expensive calculations for obtaining the lower bound (iii) of the penalty parameter, ρ_k is defined by (8), since

$$d_{k-1}^T A_{k-1} u_{k-1} = d_{k-1}^T B_{k-1} d_{k-1} + d_{k-1}^T \nabla f(x_{k-1}). \tag{27}$$

and all inner products are previously computed in the algorithm. Furthermore, the lower bound in (iii) does not depend on d_k, u_k, or ρ_k, which implies that the loop between Step 3) and Step 1) of Algorithm (3.1) is finite.

(4.3) Corollary: *The loop between Step 3) and Step 1) of Algorithm (3.1) is finite.*

To show that the line search of Step 3) of Algorithm (3.1) defines a finite sub-iteration, we use the following estimate for $\alpha_{k,i}$:

(4.4) Lemma: *Let k denote the k-th iteration of Algorithm (3.1) and assume that $\varphi_k'(0) < 0$. Then*

$$\alpha_{k,i+1} \leq \max\left(\beta, \frac{1}{2(1-\mu)}\right)\alpha_{k,i} \tag{28}$$

whenever (19) is not valid for some $i \geq 0$.

Proof: From $\varphi_k{}'(0) < 0$ and the violation of (19), we obtain

$$\begin{aligned}\bar{\alpha}_{k,i} &= \tfrac{1}{2}\frac{\alpha_{k,i}^2\varphi_k{}'(0)}{\alpha_{k,i}\varphi_k{}'(0) - (\varphi_k(\alpha_{k,i}) - \varphi_k(0))} \\ &< \tfrac{1}{2}\frac{\alpha_{k,i}^2\varphi_k{}'(0)}{\alpha_{k,i}\varphi_k{}'(0) - \mu\alpha_{k,i}\varphi_k{}'(0)} \\ &= \frac{1}{2(1-\mu)}\alpha_{k,i},\end{aligned}$$

cf. (20), and

$$\alpha_{k,i+1} \le \max\left(\beta, \frac{1}{2(1-\mu)}\right)\alpha_{k,i}.$$

∎

Since $\alpha_{k,i} \to 0$ for $i \to \infty$ and $\varphi_k{}'(0) < 0$ is impossible without violating (19), we get:

(4.5) Corollary: *The line search procedure of Step 3) of Algorithm (3.1) is finite provided that $\varphi_k{}'(0) < 0$.*

Now we are able to prove the following convergence theorem:

(4.6) Theorem: *Let x_k, v_k, d_k, δ_k, u_k, B_k, r_k, ρ_k, and J_k^* be given iterates of Algorithm (3.1), $k \ge 0$, and assume that there are positive constants γ and $\bar{\delta}$ with*

(i) $d_k^T B_k d_k \ge \gamma \| d_k \|^2$ *for all k,*

(ii) $\delta_k \le \bar{\delta}$ *for all k,*

(iii) $\rho_k \ge \dfrac{\| A_k \bar{v}_k \|^2}{\gamma(1-\bar{\delta})^2}$ *for all k,*

(iv) $\{x_k\}$, $\{d_k\}$, $\{u_k\}$, *and* $\{B_k\}$ *are bounded .*

Then there exists for each $\epsilon > 0$, a k with

a) $\| d_k \| \le \epsilon$,

b) $\| R_{k+1}^{-1/2}(u_k - v_k) \| \le \epsilon$.

Proof: First note that the boundedness of $\{u_k\}$ implies the boundedness of $\{v_k\}$, since $\alpha_k \le 1$ for all k. To show a), let us assume that there is an $\epsilon > 0$ with

$$\| d_k \| > \epsilon \tag{29}$$

for all k. From the definition of r_{k+1}, $k > 0$, we obtain either $r_j^{(k+1)} = \sigma_j^{(0)} r_j^{(0)}$ or

$$\begin{aligned}r_j^{(k+1)} &\le \frac{2m(u_j^{(k^*)} - v_j^{(k^*)})^2}{(1-\delta_{k^*})d_{k^*}^T B_{k^*} d_{k^*}} \\ &\le \frac{2m(u_j^{(k^*)} - v_j^{(k^*)})^2}{(1-\bar{\delta})\gamma\epsilon^2}\end{aligned}$$

for some $k^* \leq k$, $j = 1, \ldots, m$. Since u_k and therefore also v_k are bounded, we conclude that $\{r_k\}$ remains bounded and Lemma (2.1) implies that there is some $r > 0$ with

$$\lim_{k \to \infty} r_k = r. \tag{30}$$

Now consider any iteration index k. Then

$$\begin{aligned} \phi_{r_{k+1}}(x_{k+1}, v_{k+1}) &\leq \phi_{r_{k+1}}(x_k, v_k) + \mu\alpha_k \nabla\phi_{r_{k+1}}(x_k, v_k)^T \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix} \\ &\leq \phi_{r_{k+1}}(x_k, v_k) - \tfrac{1}{4}\mu\alpha_k\gamma \parallel d_k \parallel^2 \\ &< \phi_{r_{k+1}}(x_k, v_k) - \tfrac{1}{4}\mu\gamma\epsilon^2\alpha_k, \end{aligned} \tag{31}$$

cf. Theorem (4.2). Next we have to prove that α_k cannot tend to zero. Let $k \geq 0$, and

$$z_k \doteq \begin{pmatrix} x_k \\ v_k \end{pmatrix}, \quad p_k \doteq \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}. \tag{32}$$

Since all functions defining ϕ_r are continuously differentiable, r_{k+1} is bounded, and z_k, p_k remain in a compact subset of $\Re^{n+m}$, we can find $\bar{\alpha} > 0$ with

$$\begin{aligned} &| \nabla\phi_{r_{k+1}}(z_k + \alpha p_k)^T p_k - \nabla\phi_{r_{k+1}}(z_k)^T p_k | \\ &\quad \leq \parallel \nabla\phi_{r_{k+1}}(z_k + \alpha p_k) - \nabla\phi_{r_{k+1}}(z_k) \parallel\parallel p_k \parallel \\ &\quad \leq \tfrac{1}{8}(1 - \mu)\gamma\epsilon^2 \end{aligned} \tag{33}$$

for all $\alpha \leq \bar{\alpha}$ and for all k. Using the mean value theorem, we obtain a $\xi_k \in [0, 1)$ with

$$\begin{aligned} &\phi_{r_{k+1}}(z_k + \alpha p_k) - \phi_{r_{k+1}}(z_k) - \mu\alpha\nabla\phi_{r_{k+1}}(z_k)^T p_k \\ &\quad = \alpha\nabla\phi_{r_{k+1}}(z_k + \xi_k\alpha p_k)^T p_k - \mu\alpha\nabla\phi_{r_{k+1}}(z_k)^T p_k \\ &\quad \leq \alpha\nabla\phi_{r_{k+1}}(z_k)^T p_k + \tfrac{1}{8}\alpha(1 - \mu)\gamma\epsilon^2 - \mu\alpha\nabla\phi_{r_{k+1}}(z_k)^T p_k && \text{cf.(33)} \\ &\quad \leq -\tfrac{1}{4}\alpha(1 - \mu)\gamma \parallel d_k \parallel^2 + \tfrac{1}{8}\alpha(1 - \mu)\gamma\epsilon^2 && \text{cf.Th.(4.2)} \\ &\quad < -\alpha(\tfrac{1}{4}(1 - \mu)\gamma\epsilon^2 - \tfrac{1}{8}(1 - \mu)\gamma\epsilon^2) && \text{cf.(29)} \\ &\quad = -\tfrac{1}{8}\alpha(1 - \mu)\gamma\epsilon^2 \\ &\quad < 0 \end{aligned}$$

for all k and all $\alpha \leq \bar{\alpha}$. From (28) we conclude

$$\beta^i \leq \alpha_{k,i} \leq \bar{\beta}^i$$

for all $i \geq 0$, where

$$\bar{\beta} \doteq \max\left(\beta, \frac{1}{2(1 - \mu)}\right), \quad 0 < \bar{\beta} < 1.$$

Therefore, there is an i_0 independent from k so that (19) is satisfied for all $i \geq i_0$ and all k. Since i_k is the first index for which $\alpha_{k,i}$ satisfies (19), we conclude that α_k does not approach zero, i.e.

$$\alpha_k = \alpha_{k,i_k} \geq \beta^{i_k} \geq \beta^{i_0}.$$

Together with (31) we obtain

$$\phi_{r_{k+1}}(x_{k+1}) < \phi_{r_{k+1}}(x_k) - 2\bar{\epsilon} \tag{34}$$

with

$$\bar{\epsilon} \doteq \tfrac{1}{16}(1-\mu)\gamma\epsilon^2\beta^{i_0}.$$

Now we consider the difference

$$\begin{aligned}
&\phi_{r_{k+2}}(x_{k+1}) - \phi_{r_{k+1}}(x_{k+1}) \\
&= -\sum_{j\in J_{k+1}} (v_j^{(k+1)} g_j(x_{k+1}) - \tfrac{1}{2} r_j^{(k+2)} g_j(x_{k+1})^2) - \tfrac{1}{2} \sum_{j\in K_{k+1}} v_j^{(k+1)^2}/r_j^{(k+2)} \\
&\quad + \sum_{j\in J_k} (v_j^{(k+1)} g_j(x_{k+1}) - \tfrac{1}{2} r_j^{(k+1)} g_j(x_{k+1})^2) + \tfrac{1}{2} \sum_{j\in K_k} v_j^{(k+1)^2}/r_j^{(k+1)}
\end{aligned}$$

with

$$J_{k+1} \doteq \{1,\ldots,m_e\} \cup \{j : m_e < j \le m,\ g_j(x_{k+1}) \le v_j^{(k+1)}/r_j^{(k+2)}\},$$

$$J_k \doteq \{1,\ldots,m_e\} \cup \{j : m_e < j \le m,\ g_j(x_{k+1}) \le v_j^{(k+1)}/r_j^{(k+1)}\},$$

and K_{k+1}, K_k are the corresponding complements. Since $r_{k+1} \to r > 0$ for $k \to \infty$, $g_j(x_{k+1})$ and v_{k+1} are bounded, we get

$$\phi_{r_{k+2}}(x_{k+1}) - \phi_{r_{k+1}}(x_{k+1}) \le \bar{\epsilon}$$

for all sufficiently large k. This leads to

$$\phi_{r_{k+2}}(x_{k+1}) \le \phi_{r_{k+1}}(x_{k+1}) + \bar{\epsilon} \le \phi_{r_{k+1}}(x_k) - \bar{\epsilon},$$

cf. (34), for all sufficiently large k and to a contradiction, since $\{\phi_{r_{k+1}}(x_k)\}$ is bounded below. This shows statement a). Statement b) follows from a), the definition of r_{k+1}, cf. (12), and the boundedness of $\{B_k\}$:

$$\begin{aligned}
\| R_{k+1}^{-1/2}(u_k - v_k) \|^2 &= \sum_{j=1}^{m} \frac{(u_j^{(k)} - v_j^{(k)})^2}{r_j^{(k+1)}} \\
&\le \sum_{j=1}^{m} \frac{(1-\delta_k) d_k^T B_k d_k}{2m} \\
&\le \tfrac{1}{2} d_k^T B_k d_k.
\end{aligned}$$

∎

Note that Theorem (4.6) also treats the case in which the penalty parameters are unbounded. In that case, the convergence analysis is simplified, since definition (12) of the penalty parameters and the boundedness of $\{u_k\}$, $\{v_k\}$ imply that $\{d_k\}$ approaches zero. If, on the other hand, we knew that the penalty parameters are bounded, then (12) shows that the statement

$$\| u_k - v_k \| \le \epsilon$$

could be added to the results of Theorem (4.5).

Most of the technique in the convergence proof of Theorem (4.6) is standard and well known from unconstrained optimisation theory. It is repeated here for completeness. However, we must be aware that

$$\phi_{r_{k+2}}(x_{k+1}) > \phi_{r_{k+1}}(x_{k+1})$$

is possible, implying that convergent penalty parameters are required to obtain a contradiction to (29).

The statements of Theorem (4.6) can be used to show the approximation of a Kuhn-Tucker point by Algorithm (3.1):

(4.7) **Theorem:** *Let x_k, v_k, d_k, δ_k, u_k, B_k, J_k^* be computed by Algorithm (3.1) and assume that all assumptions of Theorem (4.6) are valid. Then there exists an accumulation point (x^*, u^*) of (x_k, u_k) satisfying the Kuhn-Tucker conditions (3) for problem (1).*

Proof: The boundedness of $\{x_k\}$, $\{u_k\}$ and the results of Theorem (4.6) guarantee the existence of $x^* \in \Re^n$, $u^* \in \Re^m$, and an infinite subset $S \subseteq \aleph$ with

$$\begin{aligned} &\lim_{k\in S} x_k = x^*, \\ &\lim_{k\in S} u_k = u^*, \\ &\lim_{k\in S} d_k = 0, \\ &\lim_{k\in S} \| R_{k+1}^{-1/2}(u_k - v_k) \| = 0. \end{aligned} \tag{35}$$

Since $\{\delta_k\}$ is bounded away from unity, (22) and (21c,d) give

$$g_j(x^*) = 0, \quad j = 1, \ldots, m_e,$$

$$g_j(x^*) \geq 0, \quad j = m_e + 1, \ldots, m,$$

showing that x^* is feasible. From (21f) we get

$$u_j^* \geq 0, \quad j = m_e + 1, \ldots, m,$$

and (21i) leads to

$$u_j^* g_j(x^*) = 0, \quad j = 1, \ldots, m. \tag{36}$$

It remains to prove (3a). Assume now there exists a $j > m_e$ so that $j \in K_k^*$ for infinitely many $k \in S$ (otherwise we are finished). The definition of K_k^*, cf. (6), implies $g_j(x^*) > \epsilon$ and (36) gives $u_j^* = 0$. We conclude from (21a) that

$$\nabla_x L(x^*, u^*) = 0.$$

∎

The following corollary follows directly from the statements of Theorems (4.6), (4.7) and from (21b).

(4.8) Corollary: *Under the assumptions of Theorem (4.7), let S define an infinite subset of $\aleph$ so that (x_k, u_k) converge to a Kuhn-Tucker point (x^*, u^*) of (1) for all $k \in S$. Then*
a) $\lim_{k \in S} \delta_k = 0$.
b) *If, in addition, the penalty parameters r_k are bounded, then*

$$\lim_{k \in S} v_k = u^*.$$

Assumptions (i) to (iii) of Theorem (4.6) are required to obtain descent directions for the function ϕ_r. As noted in the beginning of this section, the boundedness of $\{x_k\}$ and $\{d_k\}$ can be enforced by introducing additional bound constraints of the type (23), and sufficient conditions for $\{u_k\}$ to remain bounded, are given in [13]. They are mainly based on a constraint qualification which must be satisfied in each subproblem. The assumptions of the convergence theorems presented so far exclude the special case that a search direction has been obtained by (18). It can be expected that this replacement occurs rarely if the nonlinear programming problem (1) satisfies a constraint qualification at its optimal solution. If, on the other hand, (18) is always used to define the new iterates, then (3.1) could be considered as a multiplier method and its well-known convergence results can be applied.

5. Further comments

In addition to the global convergence behavior outlined in the previous section, one could be interested in the local convergence speed of Algorithm (3.1). The only statement to show is, that the steplength is one in a neighbourhood of the solution. Then (3.1) is identical with the original method of Han and Powell and we can apply their local superlinear convergence results, cf. [5] or [11], respectively. However, Algorithm (3.1) is closely related to the method presented in [13]. The only difference influencing the local convergence analysis is a slightly simplified choice of the penalty parameters. Since both approaches are identical in principle, a repetition of the local convergence analysis of [13] for the presented modified case is omitted.

Algorithm (3.1) has been implemented in a user oriented way and has been tested extensively. The usage of the program and its FORTRAN source are published in [15]. The numerical results of [15] are obtained by executing the test problems published in Hock and Schittkowski [7], and can be compared with the results given there. The subproblems of the kind (5) or (7), respectively, are solved by the quadratic programming routine of Gill, Murray, Saunders and Wright [4] and by a linear least-squares program based on the subroutines published in Lawson and Hanson [8]. Furthermore, the L_1-penalty function has been implemented to compare both approaches, and two different line search algorithms are tested.

For further information about the numerical performance of other versions of Algorithm (3.1), the reader is referred to [14]. Five different versions of the method of Wilson, Han, and Powell are tested there which all realize the active set strategy and are based on a least-squares formulation of the quadratic subproblem. They differ in the choice of the line search function, the formulation of the subproblem, the solution method for the least-squares subproblem, and in the way in which the gradients are computed. Furthermore, their performance can be compared with the performance of the 26 optimization programs tested in [12], and, in particular, with the original implementation VF02AD of Powell and with OPRQP/XROP, two versions of Bigg's [1] recursive quadratic programming method which uses an active set strategy to define equality constrained quadratic programming subproblems.

References

[1] Biggs, M.C.: *On the convergence of some constrained minimisation algorithms based on recursive quadratic programming*. J. Inst. Math. Appl. 21, 67-82 (1978)

[2] Chamberlain, R.M.: *Some examples of cycling in variable metric methods for constrained minimisation*. Math. Programming 16, 378-384 (1979)

[3] Gill, P.E., Murray, W., Wright, M.H.: *Practical optimisation*. London, New York: Academic Press, 1981

[4] Gill, P.,Murray, W., Saunders, M., Wright, M.: *The design and implementation of a quadratic programming algorithm*. Report SOL 82-1, Systems Optimisation Laboratory, Stanford University, Stanford, 1982

[5] Han, S.-P.: *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*. Math. Programming 11, 263-282 (1976)

[6] Han, S.-P.: *A globally convergent method for nonlinear programming*. J. Optimization Theory Appl. 22, 297-309 (1977)

[7] Hock, W., Schittkowski, K.: *Test examples for nonlinear programming codes*. Lecture Notes in Economics and Mathematical Systems. Vol. 187. Berlin, Heidelberg, New York: Springer, 1981

[8] Lawson, C.L., Hanson R.J.: *Solving least squares problems*. Englewood Cliffs, New Jersey: Prentice Hall, 1974

[9] Maratos, N.: *Exact penalty function algorithms for finite dimensional and control optimisation problems*. Ph.D. Thesis, Imperial College, London, 1978

[10] Powell, M.J.D.: *A fast algorithm for nonlinearly constrained optimisation calculations*. In: *Numerical analysis*. Proceedings of the Biennial Conference Held at Dundee, June 1977 (G.A. Watson, ed.) Lecture Notes in Mathematics, Vol. 630. Berlin, Heidelberg, New York: Springer, 1978

[11] Powell, M.J.D.: *The convergence of variable metric methods for nonlinearly constrained optimisation calculations*. in: *Nonlinear programming 3*. (O.L. Mangasarian, R.R. Meyer, S.M. Robinson, eds.) New York, San Francisco, London: Academic Press, 1978

[12] Schittkowski, K.: *Nonlinear programming codes. Information, tests, performance*. Lecture Notes in Economics and Mathematical Systems, Vol. 183, Berlin, Heidelberg, New York: Springer, 1980

[13] Schittkowski, K.: *The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. Part 1: Convergence Analysis*. Numer. Math. 38, 83-114 (1981)

[14] Schittkowski, K.: *The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. Part 2: An efficient implementation with linear least squares subproblems*. Numer. Math. 38, 115-127 (1981)

[15] Schittkowski, K.: *Design, implementation, and test of a nonlinear programming algorithm*.

To appear

[16] Wilson, R.B.: *A simplicial algorithm for concave programming*. Ph.D. Dissertation, Graduate School of Business Administration, Harward University, Boston, 1963

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 82-4	2. GOVT ACCESSION NO. AD-A115 667	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On the Convergence of A Sequential Quadratic Programming Method With An Augmented Lagrangian Line Search Functions.		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Klaus Schittkowski		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0267 DAAG29-81-K-0156
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research - SOL Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-143
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research - Dept. of the Navy 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE January, 1982
		13. NUMBER OF PAGES 22 pp.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Optimization; Nonlinear constraints;
Sequential Quadratic Programming Algorithms;
Superlinear convergence.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE REVERSE SIDE

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOL 82-4: ON THE CONVERGENCE OF A SEQUENTIAL QUADRATIC PROGRAMMING METHOD WITH AN AUGMENTED LAGRANGIAN LINE SEARCH FUNCTION; Klaus Schittkowski (January 1982, 22 pp)

Sequential quadratic programming methods as developed by Wilson, Han, and Powell have gained considerable attention in the last few years mainly because of their outstanding numerical performance. Although the theoretical convergence aspects of this method and its various modifications have been investigated in the literature, there still remain some open questions which will be treated in this paper. The convergence theory to be presented, takes into account the additional variable introduced in the quadratic programming subproblem to avoid inconsistency, the one-dimensional minimization procedure, and, in particular, an "active set" strategy to avoid the recalculation of unnecessary gradients. This paper also contains a detailed mathematical description of a nonlinear programming algorithm which has been implemented by the author. The usage of the code and detailed numerical test results are presented in (15).